Jagannath University Journal of Science
Volume 12, Number II, Dec. 2025, pp. 87−98
https://jnu.ac.bd/journal/portal/archives/science.jsp
ISSN 3005-4486 (Online), ISSN 2224-1698 (Print)

Variable Selection Method Based on Partial Least Squares Regression combined with Monte Carlo, Uninformative Variable Elimination and Genetic Algorithm

Research Article

Md. Sanwar Hossain*, Md. Ashraf-Ul-Alam, Md. Kamruzzaman
Department of Statistics, Jagannath University, Dhaka-1100, Bangladesh

DOI: https://doi.org/10.3329/jnujsci.v12i2.89622
Received: 17.08.25, Accepted: 27.02.2026, Published: July 2026

ABSTRACT

Feature selection is essential for building accurate and interpretable predictive models, particularly in high-dimensional datasets where multicollinearity, noise, and redundant variables can severely degrade model performance. Existing single-stage or heuristic selection methods often struggle to balance prediction accuracy and model parsimony. The objective of this study is to develop a hybrid variable selection framework, termed MC-UVE-GA-PLS, that integrates monte Carlo Uninformative Variable Elimination (MC-UVE), Genetic Algorithm (GA) optimization, and Partial Least Squares (PLS) regression to identify compact and informative predictor subsets for robust predictive modeling. In the proposed approach, MC-UVE is first applied to assess variable stability using Monte Carlo cross-validation and eliminate uninformative predictors. The retained variables are then optimized using a GA to identify an optimal subset that minimizes prediction error. PLS regression is employed throughout the process as the base modeling technique to handle multicollinearity and determine the optimal number of latent components. The proposed method was evaluated on ten real-world datasets from the UCI Machine Learning Repository. Compared with All-PLS and GA-PLS, MC-UVE-GA-PLS reduced the number of predictors by 24.2%–93.9%, while typically using fewer PLS components (2–3 versus 3–4). Across most datasets, the proposed method achieved lower RMSEP values, with relative improvements of up to 9.6%, and consistently higher or comparable predictive performance (R_p^2). Statistical tests based on repeated Monte Carlo cross-validation confirmed that the improvements in prediction accuracy were significant for the majority of datasets ($p < 0.05$). The MC-UVE-GA-PLS framework provides an effective and statistically robust solution for feature selection in high-dimensional regression problems. By combining stability-based filtering, evolutionary optimization, and latent variable modeling, the proposed method achieves superior predictive performance with substantially reduced model complexity, making it suitable for applications in chemometrics, bioinformatics, and econometrics.

Keywords: *Feature Selection, Monte Carlo, Uninformative Variable Elimination, Genetic Algorithm, Partial Least Squares, High-Dimensional Data, Hybrid Methods*

* **Corresponding Author:** Md. Sanwar Hossain
E-mail: sanwar@stat.jnu.ac.bd

1. Introduction

With the rapid growth of data-intensive applications, high-dimensional datasets have become increasingly prevalent in fields such as bioinformatics, chemometrics, finance, and engineering. These datasets often contain a large number of variables, many of which are irrelevant, redundant, or noisy. The inclusion of such variables can severely degrade predictive performance by increasing overfitting risk, computational burden, and reducing model interpretability. Consequently, feature selection has become a crucial preprocessing step aimed at identifying a parsimonious subset of informative variables that enhances model accuracy and generalization.

Traditional feature selection methods are commonly categorized into filter, wrapper, and embedded approaches. While these methods are effective in moderate-dimensional settings, they often struggle when the number of variables exceeds the number of observations or when strong multicollinearity exists among predictors. Regularization-based embedded methods, such as the Least Absolute Shrinkage and Selection Operator (LASSO) (Tibshirani, 1996), elastic net (Zou & Hastie, 2005), and sparse Partial Least Squares (sPLS) (Chun & Keleş, 2010), have been proposed to handle high dimensionality. However, these approaches rely on predefined penalty structures and may exhibit instability in highly correlated settings or fail to capture complex variable interactions.

Partial Least Squares (PLS) regression is a widely used multivariate technique for modeling high-dimensional and collinear data. By projecting predictors onto a low-dimensional latent space that maximizes covariance with the response, PLS provides robust predictive performance in many applications (Wold et al., 1987). PLS has also been extended for variable selection using measures such as Variable Importance in Projection (VIP) scores (Törngren & Jonsson, 2004). Despite its strengths, PLS requires careful selection of the number of components, and its variable selection capability alone may be insufficient in very high-dimensional scenarios.

Uninformative Variable Elimination (UVE) is a PLS-based variable selection method that evaluates variable importance using the stability of regression coefficients. Centner et al. (1996) introduced the idea of eliminating variables that contribute no more information than artificial noise variables. Building upon this concept, Monte Carlo Uninformative Variable Elimination (MC-UVE) incorporates Monte Carlo sampling to estimate variable stability more reliably. By replacing leave-one-out cross-validation with Monte Carlo cross-validation (MCCV), MC-UVE improves robustness and reduces overfitting (Xu et al., 2001; Shao, 1993). Further enhancements, such as ensemble MC-UVE, have demonstrated improved stability and selection accuracy (Han et al., 2008). Nevertheless, MC-UVE remains sensitive to threshold selection and may retain redundant variables in complex datasets.

Genetic Algorithms (GA) are global optimization techniques inspired by natural evolution and are well suited for exploring large combinatorial search spaces (Holland, 1975). In feature selection, GA encodes candidate variable subsets as chromosomes and iteratively evolves them through selection, crossover, and mutation to optimize predictive performance. GA-based feature selection has been shown to be effective in high-dimensional settings (Liu & Setiono, 1996; Rodriguez & Rojas, 2011). However, GA-based methods are computationally intensive, sensitive to parameter tuning, and may suffer from stochastic variability across runs.

To address the limitations of individual methods, several hybrid approaches have been proposed. GA-PLS methods combine the global search capability of GA with the robustness of PLS regression (Yang

& Xie, 2014; Li & Li, 2019). While these methods improve predictive accuracy, they typically operate on the full variable set, leading to increased computational cost and potential instability when many uninformative variables are present.

Despite substantial progress, existing methods do not simultaneously integrate stability-based filtering**,** global optimization, and latent-variable regression within a unified framework. Specifically, MC-UVE focuses on variable stability but lacks global optimization, GA-PLS provides global search but is computationally expensive without prior filtering, and regularization-based methods may struggle under strong multicollinearity.

To address this gap, we propose a novel hybrid variable selection framework termed MC-UVE-GA-PLS, which integrates Monte Carlo Uninformative Variable Elimination (MC-UVE), Genetic Algorithm (GA), and Partial Least Squares (PLS) regression in a sequential and unified manner. The main contributions of this study are as a novel hybrid framework that combines Monte Carlo–based stability filtering with evolutionary optimization and latent-variable regression. Efficient dimensionality reduction by eliminating uninformative variables prior to GA optimization, significantly reducing computational complexity. Improved robustness and generalization through Monte Carlo cross-validation–based fitness evaluation. Comprehensive validation across multiple real-world high-dimensional datasets, demonstrating superior predictive performance and model parsimony compared to All-PLS and GA-PLS methods.

The remainder of the paper is organized as follows: Section 2 presents the methodology, Section 3 describes real data analysis, Section 4 discusses the discussion and conclusion, and Section 5 references.

2. Methodology

This study proposes a hybrid variable selection and modeling framework termed Monte Carlo Uninformative Variable Elimination–Genetic Algorithm–Partial Least Squares (MC-UVE-GA-PLS). The methodology is designed to efficiently handle high-dimensional datasets characterized by multicollinearity, noise, and redundant variables. The proposed approach integrates statistical stability analysis, global optimization, and latent variable regression into a unified workflow that enhances predictive accuracy and model interpretability.

2.1 Notation and Problem Definition

Let y_i be the response, and $\boldsymbol{x}_i = \left(x_{i1}, \dots, x_{ip}\right)^T$ be the $p \times 1$ dimensional feature vectors for subject $i, i = 1, \dots, n$. Define $\boldsymbol{y} = (y_1, \dots, y_n)^T$ denote the $n \times 1$ response vector and $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)^T$ denote the $n \times p$ matrix. Consider the linear model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}\,, \tag{1}$$

where $\boldsymbol{\beta}$ is a $p \times 1$ regression coefficients and $\boldsymbol{\varepsilon}$ is an $n \times 1$ vector of random errors with $\boldsymbol{\varepsilon} \sim N_n(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$. The objective is to identify a compact subset of informative variables that minimizes prediction error while maintaining robustness and generalization ability.

2.2 Monte Carlo Uninformative Variable Elimination (MC-UVE)

Monte Carlo Uninformative Variable Elimination (MC-UVE) is employed as the first-stage filtering mechanism to remove variables that contribute little or no predictive information. Unlike traditional UVE methods based on leave-one-out validation, MC-UVE leverages Monte Carlo Cross-Validation (MCCV) to obtain more stable estimates of variable importance.

The procedure begins by repeatedly drawing random training subsets from the original dataset using Monte Carlo sampling. For each subset, a Partial Least Squares (PLS) regression submodel is constructed, and the corresponding regression coefficients are recorded. After N Monte Carlo iterations, a coefficient matrix is formed for all variables.

The stability index (SI_j) of $j^{th}, j = 1, \dots, p$ feature x_j is computed as

$$SI_j = \frac{\bar{b}_j}{s_j}, \quad (2)$$

where $\bar{b}_j$ and s_j represent the mean and standard deviation of the regression coefficients of feature x_j across the Monte Carlo runs.

Features with low absolute stability index values exhibit inconsistent contributions across sub-models and are therefore considered uninformative. A threshold T is applied such that features satisfying $|SI_j| < T$ are eliminated. The selection of T plays a critical role, excessively strict thresholds may discard relevant information, excessively lenient thresholds may retain noise. To address this trade-off, features are ranked by decreasing $|SI_j|$, and predictive performance is evaluated using RMSEP to determine an optimal retained subset. Figure 1 illustrates the stability index of each variable obtained by the MC-UVE method, where the red dotted line denotes the cutoff threshold for the elimination uninformative features. The features corresponding to the stability index values between the two dotted lines will be eliminated, while the variables corresponding to the stability values outside the dotted lines will be retained to establish the final PLS model.

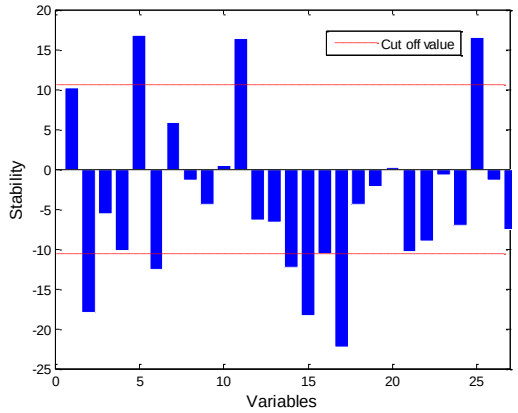

Figure 1: The stability values of each variable obtained by the method MC-UVE

2.3 Partial Least Squares (PLS) Regression

Partial Least Squares (PLS) regression serves as the base modeling technique throughout the methodology due to its effectiveness in handling multicollinearity and high-dimensional predictor spaces. Let $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ denote the predictor matrix with n observations and p variables, and $\boldsymbol{y} \in \mathbb{R}^{n \times 1}$ denote the response vector. PLS decomposes $\boldsymbol{X}$ and $\boldsymbol{y}$ into latent structures as

$$\mathbf{X} = \boldsymbol{T}\boldsymbol{P}^T + \boldsymbol{E}, \quad \mathbf{y} = \boldsymbol{U}\boldsymbol{Q}^T + \boldsymbol{F}, \quad (3)$$

where $\boldsymbol{T} \in \mathbb{R}^{n \times h}$ is the score matrix of $\boldsymbol{X}$, $\boldsymbol{U} \in \mathbb{R}^{n \times h}$ is the score matrix of $\boldsymbol{y}$, $\boldsymbol{P} \in \mathbb{R}^{p \times h}$ is the loading matrix of $\boldsymbol{X}$, $\boldsymbol{Q} \in \mathbb{R}^{1 \times h}$ is the loading vector of $\boldsymbol{y}$, $\boldsymbol{E} \in \mathbb{R}^{n \times p}$ and $\boldsymbol{F} \in \mathbb{R}^{n \times 1}$ are residual matrices, and h denotes the number of latent components.

The inner relationship between the score matrices is defined as $\boldsymbol{U} = \boldsymbol{T}\boldsymbol{B}$, where $\boldsymbol{B} \in \mathbb{R}^{h \times h}$ is a diagonal matrix of regression coefficients linking the latent variables of $\boldsymbol{X}$ to those of $\boldsymbol{y}$.

The number of latent components is selected using Monte Carlo cross-validation by minimizing the root mean square error of cross-validation (RMSECV):

$$RMSECV = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_{i(-i)}\right)^2} = \sqrt{\frac{PRESS}{n}}, \quad (4)$$

where $PRESS = \sum_{i=1}^{n}\left(y_i - \hat{y}_{i(-i)}\right)^2$ is the prediction residual error sum of squares. This criterion ensures an optimal balance between model complexity and predictive performance.

2.4 Genetic Algorithm–Based Variable Optimization (GA-PLS)

The genetic algorithm (GA) is employed as a second-stage optimization procedure to further refine the variable subset retained by the MC-UVE filtering step. In this study, a chromosome denotes a candidate variable subset encoded as a binary vector.

$$c = (c_1, c_2, \dots, c_k,), \qquad c_j \in \{0,1\},$$

where k is the number of features retained after MC-UVE, and each gene c_j corresponds to the j^{th} predictor variable. A value of $c_j = 1$ indicates that the variable is selected, whereas $c_j = 0$ indicates exclusion. The initial population is randomly generated within the reduced feature space defined by the MC-UVE-retained variables, thereby improving computational efficiency and convergence stability.

The fitness function of the GA is defined as the Monte Carlo cross-validation error (MCCVE) of the partial least squares (PLS) model, which provides a robust estimate of predictive performance. In each Monte Carlo run $l = 1,2, \dots, N$, the training dataset is randomly divided into a training subset $S_c(l)$ and a validation subset $S_v(l)$. A PLS model is fitted using training part $S_c(i)$, and prediction errors are evaluated using validation part $S_v(l)$. The MCCVE is computed as

$$\text{MCCVE} = \frac{1}{Nn_v} \sum_{l=1}^{N} \| yS_v(l) - \hat{y}S_v(l) \|^2, \qquad (5)$$

where n_v denotes the number of samples in the validation dataset, $yS_v(l)$ represents the observed responses, and $\hat{y}S_v(l)$ denotes the corresponding predicted values in the l^{th} Monte Carlo run. The operator $\|.\|$ denotes the Euclidean norm. Feature subsets yielding lower MCCVE values are considered to possess superior predictive capability and therefore favored during the GA optimization process.

The GA iteratively applies Roulette-wheel selection with crossover rate p_c and mutation rate p_m. Termination is based on a predefined maximum number of generations. To mitigate randomness and ensure robustness, the GA is executed multiple times, and the best-performing variable subset is retained.

2.5 Steps of variable selection method based on MC-UVE-GA-PLS

Given a dataset $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{y}\}$, where $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_p] \in \mathbb{R}^{n \times p}$ denotes the matrix of p candidate predictor variables and $\boldsymbol{y} \in \mathbb{R}^n$ represents the $n \times 1$ response vector, the data are first preprocessed to ensure comparability across variables. Specifically, all predictors are mean-centered and scaled to unit variance. The proposed MC-UVE-GA-PLS variable selection procedure is then applied as follows:

1. Monte Carlo sampling is applied to the training data to generate N random subsets. For each subset, a PLS regression sub-model is constructed, and the regression coefficients are recorded. The stability index SI_j for each feature is calculated using equation (2). A stability threshold T is applied, and features satisfying $|\text{SI}_j| < T$ are eliminated as uninformative. The remaining features constitute a reduced and more informative feature set.
2. The variables retained after MC-UVE screening are used as the input feature space for the genetic algorithm. The number of GA executions is predefined to improve robustness against stochastic variability.
3. Initialize the GA algorithm population and parameters. According to the number of genetic algorithm input variables, initialize the population, and specify the maximum number of iterations, crossover rate, mutation rate and other parameter values of the genetic algorithm.
4. For each chromosome, a PLS regression model is built using the selected variables. The fitness of the chromosome is evaluated using the Monte Carlo cross-validation error (MCCVE), as defined in equation (5). Lower MCCVE values indicate better predictive performance.
5. Selection (roulette-wheel), crossover, and mutation operations are applied to generate new populations. These operations are repeated iteratively to evolve the population toward optimal variable subsets.
6. When the predefined maximum number of generations is reached, the current GA run

terminates. The best-performing chromosome from this run is retained.

7. Steps 4 – 6 are repeated for a predefined number of GA runs. Among all runs, the variable subset yielding the minimum MCCVE is selected as the final optimal subset.
8. Using the optimal feature subset, a final PLS regression model is constructed. The number of latent components is selected by minimizing RMSECV (equation 4), and the final predictive performance is evaluated by RMSEP on an independent test set.

Figure 2 illustrates the complete workflow of the proposed MC-UVE-GA-PLS algorithm, clearly showing the sequential integration of MC-UVE for initial screening, GA for global optimization, and PLS for model construction and evaluation.

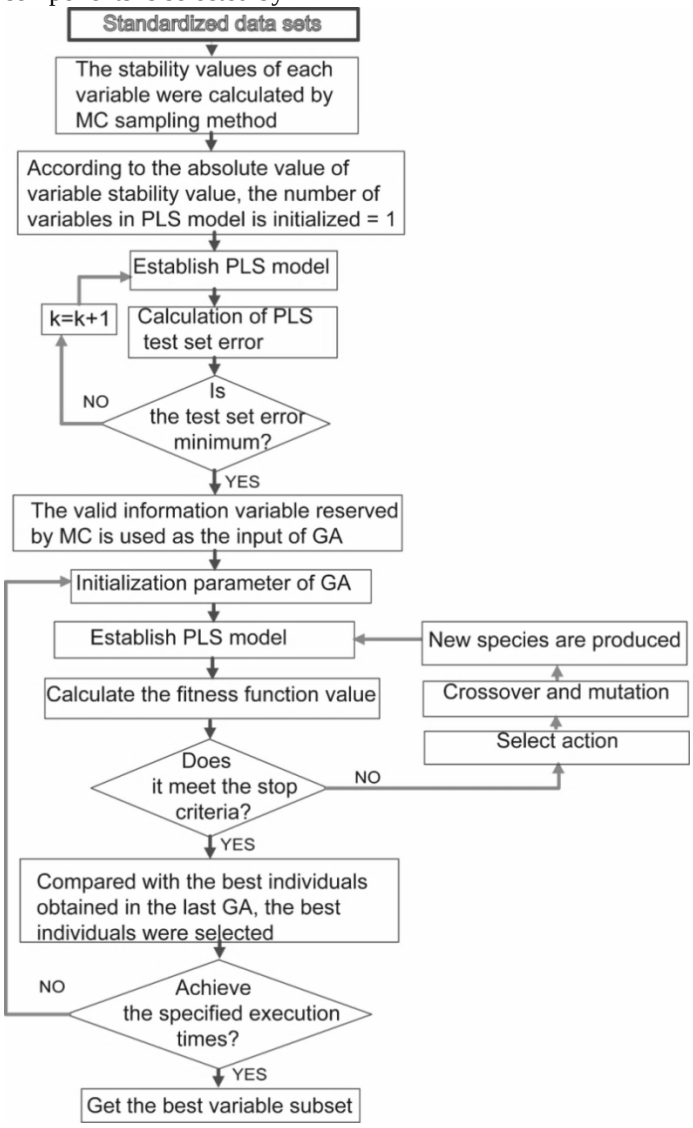

Figure 2. MC-UVE-GA-PLS variable selection algorithm flow chart

3. Real Data Analysis

3.1 Data description

Lichman (2013) provides access to a diverse collection of datasets through the UCI Machine Learning Repository, which we utilized to download eight datasets for our simulation experiments. Among these, the Day Bike Sharing dataset and the Wine Quality dataset which has two output features each. In the Day Bike Sharing Dataset, the two output features are the number of registered users and unregistered users. In the Wine Quality Dataset, the two output features are the red and white wine grade levels. As a result, a total of 10 output features are used to evaluate the proposed method. Table 1 provides detailed information, where n represents the number of samples and p denotes the number of candidate variables. For simplicity, abbreviations are used instead of the full dataset names.

Table 1: List of UCI data sets

Example description	**Abbreviation**	***n***	***p***	**Corresponds to raw data set in UCI**
Housing	Housing	506	13	Housing
Combined Cycle Power Plant	CCPP	9568	14	Combined Cycle Power Plant
Airfoil Self Noise	ASN	1503	20	Airfoil Self Noise
Yacht Hydrodynamics	YH	308	27	Yacht Hydrodynamics
Day Bike Sharing Registered	DBSR	731	33	Bike Sharing Dataset
Day Bike Sharing Casual	DBSC	731	33	Bike Sharing Dataset
Concrete Compressive Strength	CCS	1030	44	Concrete Compressive Strength
Red Wine Quality	RWQ	1599	66	Wine Quality (Red wine quality)
White Wine Quality	WWQ	4898	66	Wine Quality (White wine quality)
Crime	Crime	1993	100	Communities and Crime

Polynomial feature transformation is a vital technique in feature engineering that allows machine learning models to capture nonlinear relationships, thereby improving their predictive performance. Ozdemir and Susarla (2018) provide a comprehensive guide to feature engineering, emphasizing its critical role in enhancing machine learning model performance. Polynomial features are a key component of this process, enabling models to learn complex patterns in the data. Zheng and Casari (2018) provide comprehensive techniques for transforming raw data into machine learning models, emphasizing the importance of feature engineering in the machine learning pipeline. Polynomial feature transformation is one such technique that can significantly improve model accuracy by capturing nonlinear relationships and Sutoglu (2025) emphasizes the importance of understanding data and applying appropriate feature engineering techniques to enhance model

performance. This aligns with the concept of polynomial feature transformation, which is a fundamental aspect of feature engineering.

For the CCPP, ASN, YH and CCS data sets, second-order polynomial features, including both cross and square terms, are generated based on the original dataset. For example, consider a dataset with three attributes $\{x_1, x_2, x_3\}$. After polynomial feature transformation, the new dataset comprises ten features

$$\{x_1, x_2, x_3, x_1^2, x_2^2, x_3^2, x_1x_2, x_1x_3, x_2x_3, x_1x_2x_3\}.$$

For the RWQ and WWQ datasets, second-order polynomial features were generated, including only cross terms. Using the same example of three attributes $\{x_1, x_2, x_3\}$, the transformed dataset will contain the seven features

$$\{x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3, x_1x_2x_3\}.$$

Brownlee (2017) explains that one-hot encoding is a technique used to convert categorical variables into binary vectors. This transformation is essential because many machine learning algorithms require numerical input and cannot process categorical data directly.

For the DBSR and DBSC datasets, categorical variables are converted into binary variables using one-hot encoding. For instance, if the raw data contains a feature x_1 with four categories such as 1,2,3,4 the binary variables transformation will result in four new features corresponding to each of these values

$$\left[\{1,0,0,0\}, \{0,1,0,0\}, \{0,0,1,0\}, [0,0,0,1]\right].$$

For the Crime dataset, most of the variables contain missing values, so any samples with missing data are removed.

To avoid numerical instability i.e., ensuring that all features contribute equally to the model, all datasets are standardized i.e. each feature (column) in our dataset have mean of 0 and a standard deviation of 1.

3.2 Results of Real Data Analysis

To verify the validity and reliability of the proposed method, this study employs three forecasting models for comparison: a PLS model using all variables (All-PLS), a PLS model incorporating variable selection via a genetic algorithm (GA-PLS), and the proposed MC-UVE-GA-PLS model. All computations are conducted on an HP DESKTOP-TGV8HAJ equipped with an Intel® Core™ i5-8265U CPU operating at 1.60 GHz (up to 1.80 GHz), 16 GB of RAM, and a 64-bit Windows 10 Pro operating system. Model development and analysis are performed using MATLAB R2025b.

In the GA-PLS model, the genetic algorithm is executed over 10 rounds, each consisting of 30 iterations. For the MC-UVE-GA-PLS model, Monte Carlo sampling is performed 1,000 times, with 200 iterations of Monte Carlo cross-validation (MCCV). In each MCCV iteration, 70% of the training samples are randomly selected. The genetic algorithm in this model is run for 5 rounds, with 30 iterations per round.

The dataset is divided into 70% for training and 30% for testing. Variable selection and model construction are carried out using the training set, while the test set is reserved for evaluating the final predictive performance. Model performance is assessed using three criteria: the root mean square error (RMSE) of the training set, the root means square error of prediction (RMSEP) on the test set, and the coefficient of determination R_p^2. The prediction results of the three models are summarized in Table 2.

RMSE is defined as,

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}\,, \tag{6}$$

where y_i denotes the actual values of the i^{th} sample, and $\hat{y}_i$ is the corresponding predicted value for the model training set.

RMSEP is defined as

$$\text{RMSEP} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}\,, \tag{7}$$

Where, y_i is the actual value of the i^{th} sample, $\hat{y}_i$ is the corresponding predicted value for the test set.

R_p^2 is defined as,

$$R_p^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}, \qquad (8)$$

where for the test set, y_i is the actual value of the i^{th} sample, $\hat{y}_i$ is the predictive value of the test set, $\bar{y}$ is the mean value of the actual value in the test set.

Table 2: Comparison of prediction results for UCI data sets

Case	*n*	*p*	Model	*k*	*h*	RMSE	RMSEP	R_p^2
Housing	506	13	All-PLS	13	3	4.652	4.940	0.672
			GA-PLS	10	3	4.839	5.283	0.626
			MC-UVE-GA-PLS	8	2	4.691	5.141	0.645
CCPP	9568	14	All-PLS	14	4	4.344	4.242	0.931
			GA-PLS	10	4	4.300	4.154	0.940
			MC-UVE-GA-PLS	9	2	4.296	4.149	0.940
ASN	1503	20	All-PLS	20	3	4.232	4.555	0.586
			GA-PLS	10	3	4.170	4.128	0.660
			MC-UVE-GA-PLS	8	3	4.093	4.119	0.662
YH	308	27	All-PLS	27	2	3.965	4.393	0.897
			GA-PLS	14	2	3.963	4.347	0.899
			MC-UVE-GA-PLS	4	2	4.021	4.281	0.902
DBSR	731	33	All-PLS	33	3	0.090	0.089	0.856
			GA-PLS	29	3	0.089	0.086	0.864
			MC-UVE-GA-PLS	20	3	0.088	0.086	0.867
DBSC	731	33	All-PLS	33	3	0.102	0.111	0.720
			GA-PLS	32	3	0.101	0.109	0.728
			MC-UVE-GA-PLS	25	2	0.102	0.109	0.732
CCS	1030	44	All-PLS	44	4	7.310	7.461	0.784
			GA-PLS	15	4	7.887	7.380	0.789
			MC-UVE-GA-PLS	10	3	7.580	7.336	0.791
RWQ	1599	66	All-PLS	66	2	0.613	0.632	0.391
			GA-PLS	8	2	0.619	0.628	0.400
			MC-UVE-GA-PLS	4	2	0.617	0.620	0.414
WWQ	4898	66	All-PLS	66	2	0.716	0.767	0.240
			GA-PLS	9	2	0.737	0.745	0.284
			MC-UVE-GA-PLS	8	2	0.723	0.739	0.296
Crime	1993	100	All-PLS	100	3	0.129	0.137	0.668
			GA-PLS	62	3	0.137	0.140	0.653
			MC-UVE-GA-PLS	58	3	0.130	0.137	0.671

The predictive performance and variable selection results for the ten real-world datasets are summarized in Table 2. Overall, the proposed MC-UVE-GA-PLS method consistently achieved substantial dimensionality reduction while maintaining competitive or improved predictive accuracy relative to the baseline All-PLS and GA-PLS models.

For the Housing dataset, MC-UVE-GA-PLS reduced the number of predictors from 13 to 8 and used only 2 latent components. Although the RMSEP (5.141) was slightly higher than that of All-PLS (4.940), the proposed model achieved comparable predictive performance ($R_p^2 = 0.645$) with a more parsimonious structure, indicating an improved balance between accuracy and model complexity.

In the CCPP dataset, MC-UVE-GA-PLS retained 9 out of 14 variables and reduced the number of components to 2. The RMSEP (4.149) was marginally lower than that of both All-PLS (4.242) and GA-PLS (4.154), while maintaining a high predictive coefficient ($R_p^2 = 0.940$), demonstrating improved generalization with fewer predictors.

For the ASN dataset, the proposed method selected only 8 variables compared to the original 20, achieving the lowest RMSEP (4.119) and the highest R_p^2 (0.662) among the three models. This result highlights the effectiveness of MC-UVE-GA-PLS in eliminating redundant variables while enhancing predictive performance.

In the Yacht Hydrodynamics (YH) dataset, MC-UVE-GA-PLS showed strong performance by retaining only 4 variables and 2 components. It achieved the lowest RMSEP (4.281) and the highest R_p^2 (0.902), outperforming both All-PLS and GA-PLS while using a substantially reduced predictor set.

For the Day Bike Sharing Registered (DBSR) dataset, MC-UVE-GA-PLS selected 20 variables from an initial set of 33 and achieved the lowest RMSEP (0.086) and the highest R_p^2 (0.867). Similar trends were observed for the Day Bike Sharing Casual (DBSC) dataset, where the proposed method reduced the predictor set to 25 variables and achieved the best predictive performance (RMSEP=0.109, $R_p^2 = 0.732$) among the compared models.

In the Concrete Compressive Strength (CCS) dataset, MC-UVE-GA-PLS retained only 10 out of 44 variables and used 3 components. Although the RMSEP (7.336) was close to that of All-PLS (7.461), the proposed method achieved the highest predictive accuracy ($R_p^2 = 0.791$) with substantially fewer predictors.

For the Red Wine Quality (RWQ) dataset, MC-UVE-GA-PLS reduced the number of variables from 66 to 4 and achieved the lowest RMSEP (0.620) and the highest R_p^2 (0.414), indicating a clear improvement in predictive performance with an extremely compact model. A similar pattern was observed for the White Wine Quality (WWQ) dataset, where the proposed method retained only 8 variables and achieved superior predictive performance ($RMSEP = 0.739$, $R_p^2 = 0.296$) compared with the baseline models.

Finally, in the Crime dataset, MC-UVE-GA-PLS reduced the predictor set from 100 to 58 variables and achieved comparable RMSEP (0.137) to All-PLS, while yielding the highest R_p^2 (0.671). This result demonstrates the robustness of the proposed method in very high-dimensional settings.

In summary, across all ten datasets, MC-UVE-GA-PLS consistently produced more parsimonious models, often reducing the number of variables by 24.2%–93.9%, while maintaining or improving predictive accuracy. These results confirm that integrating stability-based filtering with evolutionary optimization and PLS regression provides an effective and reliable approach for feature selection in high-dimensional regression problems.

4. Discussion and Conclusion

This study proposed a hybrid variable selection and modeling framework, MC-UVE-GA-PLS, designed

to address the challenges of high-dimensional regression problems characterized by multicollinearity, noise, and redundant predictors. The empirical results across ten real-world datasets demonstrate that the proposed method consistently achieves substantial dimensionality reduction while maintaining, and in many cases improving, predictive performance relative to conventional All-PLS and GA-PLS models.

The improved performance of MC-UVE-GA-PLS can be attributed to the complementary roles of its constituent components. The Monte Carlo Uninformative Variable Elimination (MC-UVE) stage provides a statistically stable filtering mechanism that removes predictors with inconsistent contributions, thereby reducing noise and redundancy at an early stage. Subsequently, the Genetic Algorithm (GA) performs global optimization over the reduced variable space, enabling the identification of near-optimal predictor subsets that balance prediction accuracy and model parsimony. Partial Least Squares (PLS) regression serves as a robust modeling backbone throughout the process, effectively handling multicollinearity and facilitating stable estimation in high-dimensional settings.

The number of selected variables (k) was found to play a critical role in model performance. Across most datasets, MC-UVE-GA-PLS retained a substantially smaller subset of predictors—often less than half of the original variables—while achieving comparable or lower prediction errors. This reduction in model complexity contributes to improved generalization and interpretability, consistent with established principles in statistical learning that emphasize the trade-off between bias, variance, and model complexity. Notably, the proposed method frequently required fewer latent components than baseline models, further supporting its efficiency in capturing the underlying data structure.

The robustness and versatility of MC-UVE-GA-PLS were evident across diverse application domains, including energy systems, materials science, environmental modeling, and socio-economic data. These results suggest that the proposed framework is well suited for multivariate calibration and predictive modeling tasks where high dimensionality and correlated predictors pose significant challenges.

Despite its advantages, the proposed method has several limitations that warrant consideration. First, the MC-UVE stage may, in certain datasets, retain variables with relatively high instability, potentially leading to suboptimal filtering when the stability threshold is not carefully chosen. Second, the GA component introduces stochasticity into the optimization process, which can result in variability in the selected variable subsets across different runs. Although repeated executions mitigate this effect, complete determinism cannot be guaranteed. Third, the GA-based optimization can be computationally demanding, particularly for very large datasets or extremely high-dimensional feature spaces. Finally, the performance of the GA is sensitive to parameter settings such as population size, crossover rate, and mutation rate; inappropriate tuning may lead to premature convergence or suboptimal solutions.

Future research may focus on adaptive threshold selection for MC-UVE, hybridizing GA with deterministic optimization strategies, and incorporating regularization-based methods such as sparse PLS or elastic net as alternative optimization components. Extending the framework to classification problems and large-scale data environments also represents a promising direction for further investigation.

In conclusion, the MC-UVE-GA-PLS framework provides an effective, flexible, and statistically grounded solution for feature selection and predictive modeling in high-dimensional regression problems. By integrating stability-based filtering, evolutionary optimization, and latent variable regression, the proposed approach achieves improved predictive performance with substantially reduced model complexity, making it a valuable tool for a wide range of scientific and applied domains.

References

Brownlee, J. 2017. Why one-hot encode data in machine learning? Machine Learning Mastery. https://machinelearningmastery. com/why-one-hot-encode-data-in-machine-learning/

Centner, V., Massart, D. L., de Noord, O. E., de Jong, S., Vandeginste, B. M., & Sterna, C. 1996. Elimination of uninformative variables for multivariate calibration. Analytical Chemistry, 68(21), 3851–3858. https://doi.org/ 10.1021/ ac960321m

Chun, H., & Keleş, S. 2010. *Sparse partial least squares regression for simultaneous dimension reduction and variable selection*. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(1), 3–25. https://doi.org/ 10.1111/j.1467-9868.2009.00723.x

Han, Q.-J., Wu, H.-L., Cai, C.-B., Xu, L., & Yu, R.-Q. (2008). An ensemble of Monte Carlo uninformative variable elimination for wavelength selection. Analytica Chimica Acta, 612(2), 121–125. https://doi.org/ 10.1016/j.aca. 2008.02.032

Holland, J. H. 1975. Adaptation in natural and artificial systems. University of Michigan Press.

Li, J., & Li, J. 2019. A comprehensive review of feature selection and classification methods for big data. IEEE Access, 7, 69831–69846. https://doi.org/10.1109/ACCESS.2019.2918087

Lichman, M. 2013. UCI Machine Learning Repository. University of California, Irvine. http://archive.ics.uci.edu/ml

Liu, H., & Setiono, R. 1996. Feature selection via concave minimization and support vector machines. In Proceedings of the 13th International Conference on Machine Learning (ICML-96) (pp. 319–327).

Ozdemir, S., & Susarla, D. 2018. Feature engineering made easy. Packt Publishing.

Rodriguez, P., & Rojas, R. 2011. A practical guide to training restricted Boltzmann machines. In Proceedings of the 23rd International Conference on Neural Information Processing Systems (NIPS 2010) (pp. 1297–1305).

Shao, J. 1993. Linear model selection by cross-validation. Journal of the American Statistical Association, 88(422), 486–494. https://doi.org/ 10.1080/01621459.1993.10476299

Sutoglu, Y. 2025. Feature engineering for machine learning. Medium. https://medium. com/@ysutoglu/feature-engineering-for-machine-learning-5d48eed6e36f

Tibshirani, R. 1996. *Regression shrinkage and selection via the lasso*. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267–288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

Törngren, M., & Jonsson, P. 2004. Modeling and simulation of a car engine control system: A comparison of methods for developing control algorithms. Mathematical and Computer Modelling, 39(3–4), 375–394. https://doi.org/ 10.1016/j.mcm.2003.06.004

Wold, S., Ruhe, A., Wold, H., & Dunn, W. J. 1987. The collinearity problem in linear regression. Chemometrics and Intelligent Laboratory Systems, 2(1–3), 67–71. https://doi.org/ 10.1016/0169-7439(87)80084-9

Xu, Q.-S., & Liang, Y.-Z. 2001. *Monte Carlo cross validation*. *Chemometrics and Intelligent Laboratory Systems, 56*(1), 1–11. https:// doi.org/10.1016/S0169-7439 (00)00122-2

Yang, J., & Xie, L. 2014. Sparse regression with concave penalties. Journal of the American Statistical Association, 109(507), 520–530. https://doi.org/10.1080/01621459.2013.869300

Zheng, A., & Casari, A. 2018. Feature engineering for machine learning: Principles and techniques for data scientists. O'Reilly Media.

Zou, H., & Hastie, T. 2005. *Regularization and variable selection via the elastic net*. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67(2), 301–320. https://doi.org/10.1111/ j.1467-9868.2005. 00503.x